Remarks

The present amendment responds to the Official Action dated November 4, 2008. A petition for a one month extension of time and authorization to charge our credit card the fee of $130 is enclosed. The Official Action rejected claims 29 and 30 under 35 U.S.C. § 112, first paragraph. Claims 5, 9, and 28-30 were rejected under 35 U.S.C. § 112, second paragraph. Claims 1-5, 9-12 and 28-30 were rejected under 35 U.S.C. § 103(a) based on Lowell et al. U.S. Patent No. 3,623,017 (Lowell) in view of Kurshid U.S. Publication No. 2002/0104032 (Kurshid). Claims 6, 7, 13 and 14 were rejected under 35 U.S.C. § 103(a) based on Lowell in view of Kurshid and in view of Official Notice. Claims 15-19 and 22 were rejected under 35 U.S.C. § 103(a) based on Lowell and Kurshid in view of Ishikawa U.S. Patent No. 5,787,303 (Ishikawa).

Claims 8, 20, 21, 23-27 have been previously canceled without prejudice. Claims 1, 10, 11, 13, 15, 16, 18, 19, 22, 28, and 30 have been amended to be more clear and distinct. Claims 1-7, 9-19, 22, and 28-30 are presently pending.

Section 112 Rejections

The Official Action rejected claims 29 and 30 under 35 U.S.C. § 112, first paragraph. The Examiner suggests that there is no stage that can be described as single-cycle citing Fig. 9B of the present specification. However, Fig. 9B clearly shows PCLK CYCLE 927 as single-cycle stages for the fetch stage 928, the decode stages 929, and the execute stages 930. For example, during "the next PCLK, PCLK cycle 3 943, instruction C(3) 944 is being fetched, instruction B(2) 945 is being decoded and classified as a class 2 instruction, and instruction A(1) 946 is in execute class 1

unit 1002" as described at page 27, lines 17-19. In a similar manner, during "the next PCLK, PCLK cycle 4 950, instruction D(1) is being fetched and the pipeline advances at a class 2 four MCLK rate such that by the end of the fourth PCLK cycle 950, the instruction D(1) 951 is loaded into the IR 1010 replacing instruction C(3), the pipeline controller was notified of a pending class 3 instruction, the decoded and classified instruction C(3) 952 is loaded into the DR 1012, and the instruction B(2) 953 completes its execution in execute class 2 unit 1004 in four elapsed MCLK cycles 954-957" as described at page 28, lines 3-8. It is noted that the MCLK CYCLE 926 column of Fig. 9B illustrates how the duration of the adaptable period PCLK cycles is adapted in multiples increments of MCLK cycles depending on the instruction class. For example, see Fig. 9B where PCLK cycle 3 943 has a duration of three MCLK cycles 947-949 and PCLK cycle 4 950 has a duration of four MCLK cycles 954-957. Also, the pipeline stages shown in Fig. 10 are clocked by an adaptable PCLK clock using a "common PCLK cycle number 927" in Fig. 9B for reference, as described at page 26, line 23 – page 27, line 3 and page 27, lines 7-9 of the present specification. Fig. 9B also illustrates pipeline stages that are described as single-cycle. Thus, the rejection of claims 29 and 30 under 35 U.S.C. § 112, first paragraph should be reconsidered and withdrawn.

Claims 5, 9, and 28-30 were rejected under 35 U.S.C. § 112, second paragraph. With regard to claims 29 and 30 the Examiner suggests that it is misleading to say a stage is single-cycle when it contains multiple MCLK cycles. However, claim 29 claims a "variable duration single-cycle execution stage" which as described above addresses the use of the PCLK signal, that has a variable duration, to clock each pipeline stage as shown in Figs. 9B and 10. Thus, a "variable

duration single-cycle execution stage" is shown in Figs. 9B and 10 and described at page 27, line 7 – page 28, line 21.

Claim 30 claims a "first single-cycle time period is shorter than a second single-cycle time period". As noted above, the PCLK 927 of Fig. 9B is a single-cycle clock having, for example, PCLK cycle 3 943 as a first time period for execution of class one instructions, such as the execute cycle for class one instruction A(1) 946. Also, the PCLK 927 is shown as having, for example, PCLK cycle 4 950 as a second time period for execution of class two instructions, such as the execute cycle for class two instruction B(2) 953. The first time period is shorter than the second time period. Claims 29 and 30 are clear and supported by the specification and figures and their rejection should be reconsidered and withdrawn.

Regarding the rejection of claims 5, 9, and 28, the Examiner suggests that latency is a term associated with an instruction. To be clear, claim 1 has been amended to claim "each of the class one instructions capable of being executed in a first pipeline, <u>each stage of the first pipeline having a first execution latency</u> and each of the class two instructions capable of being executed in a second pipeline, <u>each stage of the second pipeline having a second execution latency</u>". Emphasis added.

Also, the Examiner suggests that to refer to the latency of an adaptable decode stage doesn't really make sense. The Examiner further suggests that claim language like that of claim 19 is more appropriate "wherein each stage of the normal pipeline has a duration equal to the first execution latency". However, since claim 5 already claims "wherein each stage of the normal pipeline has a duration equal to the first execution latency", which language the Examiner

suggested as being more appropriate, it is not clear what the Examiner finds inappropriate. If this rejection of claim 5 is maintained, clarification is requested. Also, since claim 9 claims "wherein the class one execution unit has the first execution latency" and "wherein the class two execution unit has the second execution latency", it is not clear what the Examiner finds inappropriate since a "latency of an adaptable decode stage" is not mentioned in claim 9. The rejection of claim 9 in light of the amendment to claim 1 from which it depends should be reconsidered and withdrawn.

The amendments to claim 28 and the base claim 1 from which claim 28 depends address the Section 112, second paragraph rejection of this claim and the rejection should be reconsidered and withdrawn.

The Art Rejections

As addressed in greater detail below, Lowell, Kurshid, Official Notice, and Ishikawa do not support the Official Action's interpretation of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicants do not acquiesce in the analysis of Lowell and Ishikawa made by the Official Action and respectfully traverse the Official Action's analysis underlying its rejections.

Claims 1-5, 9-12, and 28-30 were rejected under 35 U.S.C. § 103(a) based on Lowell in view of Kurshid. Lowell describes a processor that upon detection of an "extended sequence instruction", such as "multiply, divide, square root, etc." causes a higher speed clock to be switched in to replace a low speed clock used for normal timing of an arithmetic section. Lowell accomplishes this "by utilizing a high-speed clock when an extended sequence type of instruction

is being processed. When a normal instruction such as an add, subtract, store, transfer, etc. is being processed, the low-speed clock is operational". Lowell, col. 1, lines 5-9 and lines 20-23.

The longer execution type of instructions typically require a longer execution stage or multiple execution stages which, according to Lowell, during "the execution of this type of instruction by the arithmetic section of the computer, it is necessary to interrupt the normal timing of the computer until the extended sequence instruction has been completed." Lowell, col. 1, lines 9-12. Lowell suggests that by switching to a high speed clock for "extended sequence instructions", the interruption of normal timing may be minimized. Lowell, Abstract. Lowell further describes circuitry to switch to a high speed clock as input to the command generator "so that the command enable signals for the arithmetic section are produced at a faster rate". Lowell, col. 3, lines 19-25. To produce signals at a "faster rate" as described Lowell requires the use of multiple clock pulses to execute an extended sequence instruction. Lowell, col. 1, lines 47-55. Lowell is silent on the type of pipelining, if any, used in the arithmetic section for operation of the "extended sequence instructions". Lowell is also silent with regard to latency of pipeline stages of the processor using his invention. This complete lack of information on latency of pipeline stages demonstrates that Lowell is not concerned with changing the latency of pipeline stages. Lowell is merely interested in reducing the "arithmetic hold condition" during execution of an "extended sequence instruction". Lowell, col. 1, lines 17-19.

In the response to arguments section on page 18 of the Official Action, the Examiner responds that "it is entirely reasonable for Lowell's 'extended sequence instructions' which are distinguished based on 'a relatively long execution time' to be interpreted as 'based on an

instruction's execution latency." The Examiner's suggestion seems intended to imply that Lowell's "extended sequence instruction" is a claimed "class two instruction" and Lowell's "normal instruction" would then be a claimed "class one instruction" since, as claimed in claim 1, the "first execution latency is shorter than the second execution latency". However, amended claim 1 claims a "class two instruction capable of being executed in a second pipeline, <u>each stage of the second pipeline having a second execution latency</u>" and a "class one instruction capable of being executed in a first pipeline, <u>each stage of the first pipeline having a first execution latency</u>". (Emphasis added) As claimed, the "first execution latency" and the "second execution latency" are pipeline stage latencies not an instruction's execution latency as the Examiner suggests. Further, Lowell is silent on the type of pipelining, if any, used in the arithmetic section for operation of the "extended sequence instructions". Lowell is also silent with regard to latency of pipeline stages of the processor using his invention. Thus, Lowell does not teach and does not make obvious " an adaptable pipeline control unit responsive to the instruction class indication for adapting <u>a duration of latency of the adaptable fetch stage, the adaptable decode stage, and the adaptable execution stage</u> to the first execution latency for the class one instructions and to the second execution latency for the class two instructions dependent on the instruction class indication." as claimed in claim 1. Emphasis added.

The Official Action correctly admits that "Lowell fails to disclose that the decode and fetch unit **are adaptable based on latency.**" (emphasis added). Consequently, it is implicit that Lowell also lacks "an adaptable pipeline control unit responsive to the instruction class indication **for adapting a duration of latency of the adaptable fetch stage, the adaptable decode stage**, and

the adaptable execution stage to the first execution latency for the class one instructions and to the second execution latency for the class two instructions dependent on the instruction class indication" as claimed in claim 1, for example (emphasis added).

The Official Action's reliance upon Kurshid is not appropriate. Kurshid extends a clock "depending on the application level" and may also extend a clock based on an application level and whether an instruction being executed is a single cycle instruction. Kurshid, paragraph [0030]. Kurshid's determination whether a single cycle instruction is to be executed does not take into account the execution latency of the instruction.

It is in regard to Kurshid that the Examiner suggests that Applicant mischaracterized a definition of latency. However, this is not the case. "In developing a processor pipeline, the execution delays for each instruction, as it flows through the pipeline, are examined and typically the longest delay path through any stage sets the pipeline clock period and consequent operating frequency. The longest delay path is a worst case path and is typically referred to as a critical timing path. A pipeline stage logic path typically begins with output signals from a previous stage's storage device, such as a latch, register, or memory. The signals then flow through function specific combinatorial logic defining the functional portion of the pipeline stage. Then the pipeline stage ends with saving that stage's output in a storage device. Increasing the clock frequency reduces the clock period and typically causes the addition of pipeline stages since there is less time to accomplish the longest path function specific combinatorial logic." Present specification, page 2, lines 13-23. "Instruction timing performance is measured by each instruction's critical path, a measure of the worst-case signal propagation time for the instruction

in each pipeline stage." Present specification, page 6, lines 21-23. With regard to Fig. 1, all "pipeline stages typically advance with the same timing. The maximum clock frequency is limited by the longest latency logic path, the critical path, which corresponds to a particular instruction being processed at some stage. Often, this path will occur during the execute stage. ... In order to describe the basic inventive concepts, it is not unreasonable to assume that the longest latency path is in the execute stage for a processor with complex instructions" as described at page 9, line 24 – page 10, line 7 of the present specification. (Emphasis added) Further, for example, "Fig. 3 illustrates a graph 300 for a set of instructions which is divided into three classes: class 1 instructions 310 with an execution logic latency less than or equal to 5ns and a clock frequency of 200MHz, class 2 instructions 330 with an execution logic latency greater than 5ns but less than or equal to 6.67ns and a clock frequency of 150MHz, and class 3 instructions 320 with an execution logic latency greater than 6.67ns but less than or equal to 10ns and a clock frequency of 100MHz. (Emphasis added) Thus, as would be understood by one skilled in the art, the execution latency of an instruction "is measured by each instruction's critical path, a measure of the worst-case signal propagation time for the instruction in each pipeline stage" as described at page 6, lines 18-23 of the present specification.

Kurshid does not determine "each of the class one instructions capable of being executed in a first pipeline, each stage of the first pipeline having a first execution latency and each of the class two instructions capable of being executed in a second pipeline, each stage of the second pipeline having a second execution latency, wherein the first execution latency is shorter than the second execution latency" as claimed in claim 1 and as defined in the specification. Kurshid's "different

technique of adapting the functional units clock . . . based on instruction latency" relied upon by the Official Action is inappropriate and does not meet the terms of the claimed "adaptable decode stage for classifying and decoding the instruction stored in the instruction register and generating an instruction class indication" where the instruction class indication represents the execution latency or critical path associated with the instruction. Consequently, Kurshid cannot adapt "a duration of latency of the adaptable fetch stage, the adaptable decode stage, and the adaptable execution stage to the first execution latency for the class one instructions and to the second execution latency for the class two instructions dependent on the instruction class indication" as claimed. The Official Action fails to address the "adaptable pipeline control unit" as amended and thus fails to make a prima facie case of obviousness.

Regarding claim 30, the Examiner apparently rejected claim 30 for the same reasons used for the rejection of claim 1. In this respect, the Examiner merely cited sections of Lowell and Kurshid without any explanation as to how they applied to claim 30. In the interest of moving prosecution forward, claim 30 has been amended in a similar manner to the amendments made to claim 1. Claim 30 distinguishes from Lowell and Kurshid in a similar manner to claim 1 and is in order for allowance.

The Official Action rejected claims 6, 7, 13, and 14 under 35 U.S.C. § 103(a) based on Lowell and Kurshid in view of Official Notice. Dependent claims 2-5, 6, 7, 9-12, 28, and 29 depend from and contain all the limitations of claim 1, claims 2-5, 6, 7, 9-12, 28, and 29 distinguish from the references in the same manner as claim 1 and are in order for allowance.

Further, amended claim 13 recites, "an instruction class adaptable pipeline processor supporting at least two classes of instructions with a first class operable with a first latency for fetch, decode, and execute pipeline stages of the adaptable pipeline and a second class operable with a second latency for fetch, decode, and execute pipeline stages and where, the first latency is shorter than the second latency". Lowell in view of Kurshid does not teach and does not make obvious "a first plurality of instructions used in the program each operable with the first latency specified in a format encoding of each of the first plurality of instructions as class 1 instructions and with a second plurality of instructions used in the program each operable with the second latency specified in a format encoding of each of the second plurality of instructions as class 2 instructions". Lowell and Kurshid also do not teach and do not make obvious "modifying the application program by changing, where appropriate, the format encoding of a class 1 instruction to a format encoding that specifies the same functionality of the original class 1 instruction but provides a class 2 instruction indication when decoded to minimize power use while still meeting performance requirements of the application program" as presently claimed in claim 13. In more specific detail, Lowell does not teach changing the "format encoding" of an extended sequence instruction, which the Examiner asserts that the extended sequence instructions are class 1 instructions, to a "format encoding" of a normal instruction "that specifies the same functionality" of the extended sequence instruction but provides a normal instruction indication when decoded. Lowell is silent on format encodings of instructions and provides no motivation to change the format encodings of instructions.

The Examiner takes Official Notice that complex instructions can be changed to simpler microinstructions for purpose of quicker execution. However, this statement is traversed as technically incorrect and clarification is requested. A complex instruction implemented by a plurality of microinstructions may, if anything, take longer to execute than a complex instruction executing on a complex execution unit. No explanation or justification is given for the purported Official Notice. Further, the Official Notice does not meet the language of the claims as addressed above and no claim limitation is pointed out regarding this Official Notice. If this rejection is maintained, further clarification is requested. Claim 13 is not obvious and is not taught by Lowell. Kurshid and Official Notice do not cure these deficiencies.

Since dependent claim 14 depends from and contains all the limitations of claim 13, claim 14 distinguishes from the reference in the same manner as claim 13 placing claims 13 and 14 in order for allowance.

Claims 15-19 and 22 were rejected under 35 U.S.C. § 103(a) based on Lowell in view of Kurshid and Ishikawa. As addressed in detail above, Lowell does not teach and does not make obvious classifying an instruction based on the instruction's execution <u>pipeline stage latency</u>. Lowell also does not teach and does not make obvious "adapting the <u>pipeline stage latency</u> of the adaptable fetch pipeline stage, the adaptable decode pipeline stage, and the plurality of adaptable execution pipeline stages of the plurality of instruction class controllable pipelines dependent on the instruction class indication" as claimed in amended claim 15. Kurshid does not cure these deficiencies.

Ishikawa describes a VLIW architecture for adjusting individual instruction execution dependent upon operand dependency interlocks. Ishikawa describes a four issue VLIW which may contain instructions having different execution times. Ishikawa checks for operand usage between instructions in the VLIW and based on operand dependencies allows an instruction to proceed with execution or be held up from executing until an operand dependency is resolved. Ishikawa, col. 3, lines 20-36. Ishikawa does not teach and does not make obvious a VLIW processor having "executable function instructions located in multiple instruction slot format VLIWs, the class one instructions each having a first execution pipeline stage latency and the class two instructions each having a longer second execution pipeline stage latency". Ishikawa does not teach and does not make obvious "an adaptable VLIW fetch pipeline stage" or "an adaptable decode pipeline stage for ... generating an instruction class indication for each of the plurality of executable function instructions". Ishikawa also does not teach and does not make obvious "an adaptable pipeline control unit" for "adapting the pipeline stage latency of the adaptable VLIW fetch pipeline stage, the adaptable decode pipeline stage, and the plurality of adaptable execution pipeline stages of the plurality of instruction class controllable pipelines dependent on the instruction class indications", as claimed in amended claim 15. Consequently, Ishikawa does not cure the deficiencies of Lowell.

Since dependent claims 16-19 and 22 depend from and contain all the limitations of the amended claim 15, respectively, claims 16-19 and 22 distinguish from the references in the same manner as claim 15 and claims 15-19 and 22 are in order for allowance.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,

Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600

800.0118
A01560

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Barry et al.

Serial No.: 10/805,803

Filed: March 22, 2004

For: METHODS AND APPARATUS FOR VARYING PIPELINE FREQUENCY BY INSTRUCTION TYPE

Group: 2183

Examiner: Johnson, Brian P.

Durham, North Carolina
March 4, 2009

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Amendment

Sir:

In response to the Office Action of November 4, 2008, please amend the above identified application as follows: